Plenary Talk

Quantum Software Models: Instantiating Abstractions

Iaakov Exman
Software Engineering Department
The Jerusalem College of Engineering
Jerusalem, Israel

Abstract

Assuming that sooner or later quantum computing will materialize as a practical addition to classical computing, we foresee desirable features and design procedures for quantum and hybrid software systems. However, we claim that it is not reasonable to have totally distinct modular design procedures, one for classical software systems and another for quantum software systems. We argue in favor of a single unified design procedure for quantum, classical and hybrid software systems, reasoning that: (1) *Classical software systems are classical limits of quantum systems* – as usual in physics; (2) *Software Systems instantiate Abstractions* – any kind of abstractions, from numbers to higher-level conceptual systems; (3) *Quantum and Classical Software have State/Operator duality* – a quantum Density Matrix is both a state and a projection operator; a classical program is both a "readable code" state and a "runnable" operator on states. Our previous *Linear Software Models* enable formal linear algebraic procedures for modular design of classical software systems. Modularization has been performed by a spectral approach applied to matrix representations, e.g. the Laplacian of the software system. This approach clearly hints toward a single unified design procedure: (1) *Linear Algebra* – the basic objects of Linear Algebra also are the basic objects of quantum computation; (2) *Laplacian Matrix* – the Laplacian, a useful representation of classical software systems, is easily modified into a Density Matrix, a single unified representation of both classical and quantum software systems. We first point out the nature of problems that may occur while designing whole software systems involving quantum computation. Then we describe a proposal for a common design procedure starting point for both classical and quantum software systems, viz. Von Neumann's quantum notion of Density Matrix. This proposal formulates modular design in terms of projection operators obtained from a design Density Matrix. We show, in the classical case, their equivalence to the Linear Software Models results obtained from the Laplacian matrix spectrum. The application in practice of the design procedure for classical, quantum and hybrid software systems is illustrated by case studies.

About the Speaker

Prof. Iaakov Exman is a faculty member of the Software Engineering Department at The Jerusalem College of Engineering, JCE, Azrieli, in Jerusalem, Israel. He got his M.Sc. degree from the Technion Institute of Technology, in Haifa, Israel, with a thesis on "Calculations with Gaussian Functions and a Model Hamiltonian". His Ph.D. degree at the Hebrew University of Jerusalem was obtained with a thesis in the area of "Information Theory". He has done post-doctoral research at Stanford University, CA, USA, working on "Computational Drug Discovery and Design". After a long term industrial experience, in large industries in the AeroSpace area and in a small agile start-up in the Software Parallel Processing area, Prof. Exman returned to the academia to dedicate himself to Software Engineering. He has published a series of papers entitled "*Linear Software Models*" in international conferences, journals and book chapters. These papers focus on a formal and practical mathematical theory of software system design, based on algebraic structures, such as the Modularity Matrix, the Laplacian Matrix and conceptual Lattices. Recently, inspired by the Linear Software Models, a paper has been accepted for presentation in an international conference workshop, opening a new series on "*Quantum Software Models*", which is the basis of the current plenary lecture. Iaakov Exman has collaborated with research groups in Spain, Germany, Sweden and Italy, and has scientific interactions with researchers in the USA. He is an editor in the area of *Theory of Software Engineering* in the Board of the International Journal of Software Engineering and Knowledge Engineering.